Fractional Repetition Codes for Repair in Distributed Storage Systems

Salim El Rouayheb and Kannan Ramchandran
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
{salim, kannanr}@eecs.berkeley.edu

***Abstract*—We introduce a new class of *exact Minimum-Bandwidth Regenerating* (MBR) codes for distributed storage systems, characterized by a low-complexity *uncoded* repair process that can tolerate multiple node failures. These codes consist of the concatenation of two components: an outer MDS code followed by an inner repetition code. We refer to the inner code as a *Fractional Repetition* code since it consists of splitting the data of each node into several packets and storing multiple replicas of each on different nodes in the system.**

Our model for repair is *table-based*, and thus, differs from the random access model adopted in the literature. We present constructions of Fractional Repetition codes based on *regular graphs* and *Steiner systems* for a large set of system parameters. The resulting codes are guaranteed to achieve the storage capacity for random access repair. The considered model motivates a new definition of capacity for distributed storage systems, that we call *Fractional Repetition* capacity. We provide upper bounds on this capacity while a precise expression remains an open problem.

I. Introduction

Despite being formed of unreliable nodes having a short lifespan, *distributed storage systems* (DSS) are required to store data for long periods of time with a very high reliability. Typically, nodes in the system will unexpectedly leave for different reasons such as hardware failures in data centers, or peer churning in peer-to-peer (P2P) systems. To overcome this problem, a two-fold solution can be adopted based on *redundancy* and *repair* [1]. Classical erasure codes can be used to introduce redundancy in the system to protect the data from being lost when nodes fail. In addition, to maintain a targeted high reliability, the system is repaired whenever a node fails by replacing it with a new one.

A distributed storage system is formed of n storage nodes and gives the user the flexibility to recover its stored file by contacting any k out of the n nodes, for some $k < n$. We call this property the *MDS property* of the DSS in reference to Maximum Distance Separable (MDS) codes. When a node fails, the system is repaired by replacing the failed node with a new "blank" node. The new node contacts d survivor nodes, downloads encodings of their data and stores it, possibly after processing it. The data stored on the new node should maintain the MDS property of the DSS. In analogy with classical codes defined by the pair (n, k), a DSS is specified by the triplet (n, k, d), where the additional parameter d, referred to as the *repair degree*, accounts for the repair requirement. Fig. 1 depicts a $(4, 2, 3)$ DSS where node v_1 has failed and has been replaced by node v_1' that contacts $d = 3$ survivor nodes in the system to download its data.

Dimakis et al. introduced and studied in [2], [3], [4] the design of erasure codes with efficient repair capabilities, termed *regenerating codes*. The authors showed the existence of a tradeoff between storage capacity and repair bandwidth in these systems. In this tradeoff, two regimes are of special interest, the minimum-bandwidth regime and the minimum-storage regime. This original work focused on *functional repair* where the only requirement on the data regenerated at the replacement node is to maintain the MDS property of the system. Subsequent works focused on the design of *exact* regenerating codes that can repair the system by reproducing an exact replica of the lost data. Rashmi et al. presented in [5] constructions of exact minimum-bandwidth regenerating (MBR) codes for the case of $d = n - 1$, and for all feasible values of the repair degree d in [6]. The existence of exact regenerating codes for the minimum-storage regenerating (MSR) case was demonstrated in [7], [8], and deterministic constructions were investigated in [5], [9], [10], [11].

In this work we are interested in the construction of *exact* minimum-bandwidth regenerating (MBR) codes that are characterized by a low-complexity repair process. In this regime, a replacement node recovers an exact copy of the lost data by contacting d survivor nodes and downloading and storing one packet of data from each. To guarantee that the constructed codes have low complexity, we require them to satisfy what we call the *uncoded repair* property: a survivor node reads the exact amount of data he needs to send to a replacement node and forwards it without any processing. Our motivation is that in practical systems the read/write bandwidth of the storage nodes is the bottleneck since it is much smaller than the network bandwidth [12]. Regenerating codes, such as random network codes [3], do not satisfy the uncoded repair property in general since they require a survivor node to read all his stored data in order to send a linear combination of them to the replacement node. We show the surprising result that even with the two apparently restrictive constraints of exact and uncoded repair, it is possible to construct optimal regenerating codes under a table-based repair model.

Our codes are based on a generalization of the construction of Rashmi et al. in [5] and are formed by the concatenation of two component codes (see Fig. 2): an outer MDS code to ensure the required MDS property of the DSS, and an

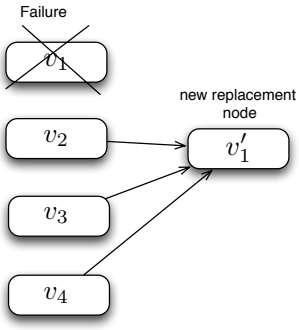

Fig. 1. An example of a distributed storage system with $(n, k, d) = (4, 2, 3)$. Initially, the system is formed of $n = 4$ nodes, $v_1, \ldots, v_4$, storing coded packets of a file. The user contacts any $k = 2$ nodes and should be able to decode the stored file. When a node fails, it is replaced by a new one that contacts $d = 3$ nodes to download its data. The figure shows an instance where node v_1 fails and is replaced by node v_1'.

inner repetition code characterized by an efficient uncoded repair process that is resilient to multiple node failures. The design of the inner code represents the challenging task in this construction. We refer to it as *Fractional Repetition (FR)* code since, in our proposed solution, the data stored on each node is split into d packets, each of which is repeated a certain number of times in the system. We study the design of FR codes that can achieve the DSS capacity for the MBR point in [3] which assumes random access repair where the new node can contact any d survivor nodes. For single failures, we provide a construction based on *regular graphs* for all feasible values of system parameters. For the general case of multiple failures, we propose code constructions based on *Steiner systems*. The table-based repair model motivates a new concept of storage capacity for distributed storage systems that we call *Fractional Repetition* (FR) capacity, which we investigate.

The rest of the paper is organized as follows. In Section II, we define our system model and motivate our code design requirements. In Section III, we give two examples of Fractional Repetition codes that are used for constructing optimal exact MBR code with uncoded repair. We describe code constructions based on regular graphs for the single failure case in Section IV. Furthermore, we provide constructions based on Steiner systems for the multiple failures case in Section V. In Section VI, we define the Fractional Repetition capacity of a DSS and provide some bounds. We conclude in Section VII with a summary of our results and discuss related open problems.

II. Motivation and Model

A distributed storage system DSS is defined by the triplet (n, k, d), where n is the total number of storage nodes in the system, $k < n$ is the number of nodes contacted by the user to retrieve his stored file, and $d \geq k$ is the repair degree that specifies the number of nodes contacted by a replacement node during repair.

We consider distributed storage systems operating in the minimum-bandwidth regime on the storage/bandwidth tradeoff curve described in [3]. Our focus on this regime is motivated by the asymmetrical cost of resources in practical systems where bandwidth is more expensive than storage. In this case, the repair bandwidth of the system, *i.e.*, the total amount of data downloaded by a replacement node is minimized. As a result, the new node needs to download only the amount of data he will store, but no more.

For load-balancing requirements, we assume a symmetric model for repair where the replacement node downloads and stores an equal amount of data, referred to as a packet, from each of the d nodes it contacts. Therefore, in the minimum bandwidth regime, d also represents the node storage capacity expressed in packets.

Under this model, the storage capacity C_{MBR} in packets of the DSS, representing the information-theoretic limit on the maximum file size that can be delivered to any user contacting k out of the n nodes, was shown in [3] to be

$$C_{MBR}(n, k, d) = kd - \binom{k}{2}. \tag{1}$$

This expression assumes a *functional repair* model where the only constraint on the data regenerated (stored) at the new node is maintaining the MDS property of the system. This allows the regenerated data to be different from the lost data as long as it is "functionally" equivalent. However, a more stringent form of repair, known as *exact* repair, that is capable of reproducing an exact copy of the lost data, may be required for many system considerations such as maintaining a systematic form of the data, reducing protocol overhead and guaranteeing data security [13], [14]. In this case, regenerating codes are also referred to as being exact. Recently, Rashmi et al. showed the interesting result that, in the minimum-bandwidth regime, there is no loss in the DSS capacity incurred by requiring exact repair, and constructed exact MBR codes that achieve the capacity C_{MBR} of (1).

Existing code constructions suffer in general from a high complexity repair process. A survivor node asked to help in repair typically has to read all his d stored packets and compute a linear combination of them in order to obtain a single packet to be forwarded to the replacement node. In addition to the computational complexity overhead, the repair process results in long delays at the survivor nodes since in general their read/write bandwidth is much smaller than the network bandwidth [12]. This motivates us to study exact MBR codes with fast and low-complexity repair where a survivor node reads only one of his stored packet and forwards it to the replacement node with no additional processing. We refer to this property as *uncoded repair*.

III. Examples

Before introducing our general constructions, we present two examples of exact regenerating codes that can achieve the capacity C_{MBR} while satisfying the uncoded repair property. The first example is based on the construction of exact MBR codes for $d = n - 1$ of Rashmi et al. in [5].

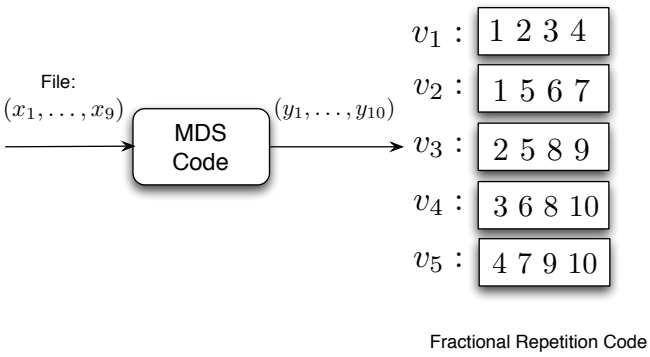

Fig. 2. An exact regenerating code for a $(5, 3, 4)$ DSS that can achieve the capacity $C_{MBR} = 9$. The code is formed by a $(10, 9)$ parity-check MDS code followed by a special repetition code. The repetition code is defined by listing the indices of the coded packets stored on each node. We refer to the inner code as a *Fractional Repetition* code of repetition degree $\rho = 2$ since the data on each node is split into $d = 4$ packets where each is repeated twice in the system. The overall code can achieve the MDS property of the DSS along with exact and uncoded repair in the case of one failure.

Example 1: Consider a $(5, 3, 4)$ DSS where the storage capacity is equal to 9 packets according to (1). Let $X = (x_1, \ldots, x_9) \in \mathbb{F}_q^9$ denote the file of 9 packets to be stored on the system, where $\mathbb{F}_q$ is the finite field of size q. Figure 2 depicts an exact MBR code that can achieve the above storage capacity [5]. This regenerating code consists of the concatenation of two components: an outer $(10, 9)$ parity-check MDS code followed by a special repetition code. The MDS code takes the file X as input and outputs the codeword $Y = (y_1, \ldots, y_{10})$, where $y_i = x_i, i = 1, \ldots, 9$, and y_{10} is a parity-check packet, *i.e.*, $y_{10} = \sum_{i=1}^{9} x_i$. The coded packets $y_1, \ldots, y_{10}$ are then placed on the 5 storage nodes following the pattern of the inner code in Fig. 2. That is, nodes $v_1, \ldots, v_5$ store, respectively, $\{y_1, y_2, y_3, y_4\}, \{y_1, y_5, y_6, y_7\}, \{y_2, y_5, y_8, y_9\}, \{y_3, y_6, y_8, y_{10}\}$ and $\{y_4, y_7, y_9, y_{10}\}$.

A user contacting a node can download all its stored packets. Since the repetition code is such that any two nodes share exactly one packet, a user contacting $k = 3$ nodes will be able to download 9 distinct packets out of the 10 coded ones (12 in total, of which 3 are repeated twice). Thus, due to the MDS property of the outer code, it can recover the whole file X. Moreover, each of the coded packets is replicated twice in the system on two different nodes which guarantees uncoded exact repair in the case of a single node failure. Indeed, whenever a node fails, its data can be recovered exactly by contacting the four surviving nodes and downloading one packet from each. For instance, when node v_1 fails, a replacement node contacts nodes $v_2, \ldots, v_5$, and downloads packets $y_1, \ldots, y_4$ from each, respectively.

The previous code is a special example of the exact MBR codes devised in [5] where uncoded repair was not a requirement. This construction, however, is limited to systems with repair degree $d = n-1$ that require contacting all the survivor nodes when a failure occurs. This may not always be feasible due, for example, to nodes having limited access bandwidth or being temporarily down. Moreover, the uncoded repair process here cannot tolerate multiple failures together, which may not be a rare event in large-scale systems where failures can also be correlated, or systems where repair is not immediate but performed at prescheduled intervals. For these reasons, we are interested in exact MBR codes for systems with small repair degree d that have an uncoded repair process that can tolerate multiple failures.

To introduce our constructions which will be detailed in the following sections, we give a second example for a DSS with $(n, k, d) = (7, 3, 3)$ having an exact and uncoded repair process that can tolerate up to two failures.

Example 2: Consider a $(7, 3, 3)$ DSS where the system storage capacity is $C_{MBR} = 6$ by (1). The code that we propose for this system is also constructed by concatenating two constituent codes: an outer $(7, 6)$ MDS code that outputs coded packets indexed from 1 to 7, followed by the repetition code depicted in Fig. 3(a). It can be seen that each of the 7 packets forming the output of the outer code is replicated on 3 different nodes in the system. Therefore, the system will always have a surviving copy of each packet in the case of two failures. Thus, the code guarantees exact uncoded repair for up to two node failures.

Next, we check that this code indeed achieves the capacity C_{MBR}. The structure of the inner repetition code is deduced from the projective plane of order 2, also called the Fano plane, depicted in Fig 3(b) [15, Chap 2]. The Fano plane consists of 7 points indexed from 1 to 7 and the following 7 lines $123, 345, 156, 147, 257, 367$ and 246, including the circle. To form the inner repetition code, we associate to each line in the Fano plane a distinct storage node in the DSS. Then, the three points belonging to that line give the indices of the packets stored on the corresponding node. In the projective plane, any two lines intersect in exactly one point. Therefore, any two nodes will have exactly one packet in common. This implies that any user that contacts 3 different nodes can get *at least* $3 \times 3 - \binom{3}{2} = 6$ distinct packets. For instance, a user contacting nodes v_2, v_4 and v_5 will get exactly 6 different packets, namely, all the packets except the one with index 6. Whereas another user contacting v_1, v_3 and v_4 will get all the 7 packets. In a worst-case analysis, the capacity of the system is limited by the user that gets the least number of packets, which is 6 here. Hence, the outer MDS code allows any user to recover the stored file of 6 packets which is exactly the capacity $C_{MBR}(7, 3, 3)$ of (1).

The previous two examples highlight the central role of the inner repetition code that allows us to obtain the desired uncoded and exact repair properties of the code in addition to achieving the capacity C_{MBR} by carefully placing the different copies of the coded packets on different nodes in the system. We call the inner code a *Fractional Repetition* (FR) code of repetition degree ρ since the content of each node is split into d packets and ρ replicas of each are stored on different nodes in the system. For instance, the inner code in the first example was an FR code with $\rho = 2$, whereas in

Original repair model in [3]	Repair model of FR codes
Functional: regenerated data should satisfy the MDS property.	*Exact*: regenerated data is an exact copy of the lost one.
Coded: new node downloads linear combination of packets.	*Uncoded*: new node downloads a specific packet with no coding.
Random-Access: the new node contacts any d surviving nodes.	*Repair Table*: a table specifies the set of d nodes to be contacted for repair.

TABLE I

A COMPARISON BETWEEN THE MODEL FOR REPAIR IN THE ORIGINAL WORK OF DIMAKIS ET AL. IN [3] AND THE MODEL FOR REPAIR FOR THE FRACTIONAL REPETITION CODES PROPOSED HERE.

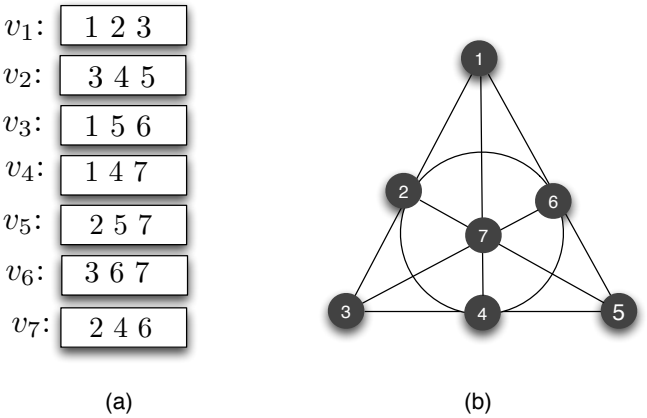

Fig. 3. (a) The inner repetition code of Example 2 for a $(7, 3, 3)$ DSS. This corresponds to a Fractional Repetition code with repetition degree $\rho = 3$. The code structure is derived from the Fano plane depicted on the right. Each of the 7 lines in the Fano plane, including the circle, corresponds to a distinct storage node. The points lying on that line give the indices of the packets stored on the node. The overall code can achieve the capacity $C_{MBR} = 6$ for this DSS, and has an exact and uncoded repair process. (b) The Projective plane of order 2, also known as the Fano plane.

the second example $\rho = 3$.

Notice that in Example 2, a replacement node has to contact a *specific* set of d nodes for repair, depending on which nodes have failed. For example, when node v_1 fails, a replacement node can recover all the lost packets by contacting nodes $\{v_4, v_5, v_6\}$, but not $\{v_2, v_3, v_4\}$. We assume that there is a *repair table* maintained in the system that is available to all the nodes in the DSS. The repair table indicates for each possible failure pattern the set of nodes that can be contacted for repair, and which packet to download from each. This table-based repair model will be adopted throughout this paper and differs from the random access model adopted in the literature where repair can be performed by contacting *any* d survivor node. We believe that this relaxation in the repair model is a well-justified price to pay in order to obtain low-complexity regenerating codes, and goes along with practical system implementations that always include a tracker server that stores the system metadata.

Table I summarizes the differences between the repair model adopted here and the original model of [3].

IV. FRACTIONAL REPETITION CODES WITH $\rho = 2$

The previous two examples suggest a general method for constructing exact MBR codes with uncoded repair process that is resilient to multiple failures. This construction consists of concatenating an outer MDS code with an inner Fractional Repetition code with repetition degree ρ that can tolerate up to $\rho - 1$ nodes failing together. Since MDS codes exist for all feasible parameters provided that the packets are taken from an alphabet of large enough size, the challenging part of the suggested construction is designing the Fractional Repetition code. Assuming all the packets in the system are to be equally protected, we are motivated to provide the following general definition of FR codes:

Definition 3 (Fractional Repetition Codes): A *Fractional Repetition* (FR) code $\mathcal{C}$, with repetition degree ρ, for an (n, k, d) DSS, is a collection $\mathcal{C}$ of n subsets $V_1, V_2, \ldots, V_n$ of a set $\Omega = \{1, \ldots, \theta\}$ and of cardinality d each, satisfying the condition that each element of Ω belongs to exactly ρ sets in the collection.

In this definition, each set V_i contains the indices of the coded packets at the output of the outer MDS code that are stored on node $v_i, i = 1, \ldots, n$. The value of θ, which will be determined later, corresponds to the length of the codewords of the outer MDS code. For instance, following this definition, the FR code of Example 2 can be written as $\mathcal{C} = \{V_1, \ldots, V_7\}$ with $V_1 = \{1, 2, 3\}, V_2 = \{3, 4, 5\}, V_3 = \{1, 5, 6\}, V_4 = \{1, 4, 7\}$ $V_5 = \{2, 5, 7\}, V_6 = \{3, 6, 7\}, V_7 = \{2, 4, 6\}$, where $\theta = 7$ and $\Omega = \{1, \ldots, 7\}$.

We focus first on the design of Fractional Repetition codes of repetition degree $\rho = 2$ with an uncoded repair that is tolerant to a single failure. We provide a code construction based on *regular graphs* that can achieve the capacity C_{MBR} of (1) for all feasible values of n and d.

To that end, we define the rate $R_{\mathcal{C}}(k)$ of an FR code $\mathcal{C}$ as the maximum file size, i.e., the maximum number of distinct packets, that the code is guaranteed to deliver to *any* user contacting k nodes.

Definition 4 (FR Code Rate): The rate $R_{\mathcal{C}}(k)$ of an FR code $\mathcal{C} = \{V_1, V_2, \ldots, V_n\}$ for a DSS with parameters (n, k, d) is defined as

$$R_{\mathcal{C}}(k) := \min_{\substack{I \subset [n] \\ |I| = k}} |\cup_{i \in I} V_i|, \tag{2}$$

with $[n] = \{1, \ldots, n\}$.

As it can be seen from the previous examples and the above definition, the DSS parameter k specifying the number of nodes contacted by a user, is not intrinsically related to the construction of the FR code. An FR code designed for a DSS with parameters (n, k_1, d) can be seamlessly used for another DSS with parameters (n, k_2, d), with $k_1 \neq k_2$. An FR code $\mathcal{C}$ is said to be *universally good* if its rate is guaranteed to be no less than the capacity C_{MBR} of the DSS,

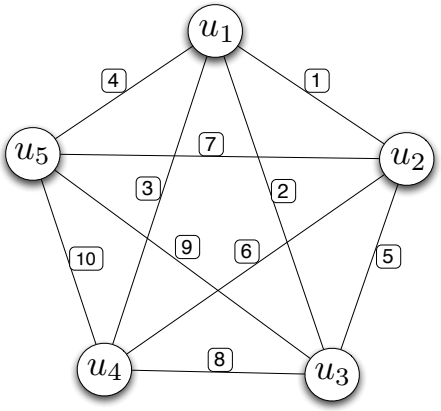

Fig. 4. The complete graph K_5 on 5 vertices. The labeling of the edges from 1 to $\binom{5}{2} = 10$ gives the FR code with $\rho = 2$ for the DSS $(5, 4, 3)$ depicted in Fig. 2. The edges adjacent to vertex u_i give the indices of the packets stored on node v_i in the DSS.

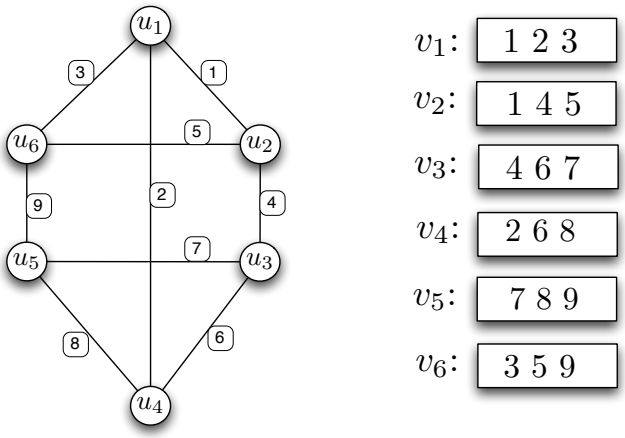

Fig. 5. (a) $R_{6,3}$ a 3-regular graph on 6 vertices. All the six vertices have constant degree equal to 3. The edges of the graph are indexed from 1 to $\frac{nd}{2} = \frac{6\times 3}{2} = 9$. (b) The corresponding universally good FR code with $\rho = 2$ obtained by Construction 1 for a DSS with $n = 6$ an $d = 3$.

i.e., $R_{\mathcal{C}}(k) \geq C_{MBR}(n, k, d)$, for all $k = 1, \ldots, d$. Here, the inequality follows from the fact that FR codes can have rates that exceed C_{MBR} due to the table-based repair relaxation, a property that will be investigated further in Section VI.

An (n, k, d) DSS stores nd packets in total. When an FR code of degree ρ is used, θ distinct packets are stored in the system, where each is replicated exactly ρ times. Therefore, the following relation exists between the FR code parameters:

Proposition 5: The parameter θ in Def. 3 of an FR code of degree ρ for an (n, k, d) DSS is given by,

$$\theta\rho = nd. \tag{3}$$

The Exact MBR codes of Rashmi et al. were proposed in [5] as capacity achieving codes for the special case of $d = n-1$. In this case, when a node fails, all the remaining nodes in the system are contacted by the replacement node, which implies that the random access and table-based repair models are equivalent.

These codes can be viewed as special FR codes with repetition degree $\rho = 2$ as shown in Example 1. Their general construction can be described with the assistance of a complete graph K_n defined on n vertices $u_1, \ldots, u_n$, with edges indexed from 1 to $\binom{n}{2}$. Prop. 5 gives $\theta = \frac{n(n-1)}{2} = \binom{n}{2}$ distinct packets. The FR code is obtained by storing on node $v_i, i = 1, \ldots, n$, the packets having the same indices as the edges adjacent to vertex u_i in K_n. Figure 4 depicts the complete graph K_5 with its edges indexed in a way to give the FR code of Fig. 2.

Next, we describe a construction of FR codes with repetition degree $\rho = 2$ and $d < n - 1$. For $\rho = 2$, Prop. 5 gives a necessary condition for the existence of FR codes, that is, nd should be even. We will show that this is also a sufficient condition and provide a general code construction based on regular graphs.

A d-regular graph $R_{n,d}$ on n vertices is a *simple* graph where all vertices have the same degree d, *i.e.*, the same number of neighboring nodes. The graph $R_{n,d}$ has $\frac{nd}{2}$ vertices, and exists whenever nd is even [16].

Construction 1: An FR code with repetition degree $\rho = 2$ can be constructed for an (n, k, d) DSS, with nd even, in the following way:

1) Generate a d-regular graph $R_{n,d}$ on n vertices $u_1, \ldots, u_n$.
2) Index the edges of $R_{n,d}$ from 1 to $\frac{nd}{2}$.
3) Store on node v_i in the DSS the packets indexed by the edges that are adjacent to vertex u_i in the graph $R_{n,d}$.

The regular graph in Step 1 can be randomly generated using efficient randomized algorithms that are well-studied in the literature, see for example [17]. The fact that the FR codes obtained by this construction have repetition degree $\rho = 2$ is a direct consequence of the graph being simple with each edge being adjacent to exactly two vertices. This also implies that any two nodes cannot have in common more than one packet. Therefore, among any k nodes observed by a user, there are at most $\binom{k}{2}$ repeated packets which corresponds to the case when any two nodes share a distinct packet. Therefore, we have the following lemma.

Lemma 6: The FR codes with repetition degree $\rho = 2$ obtained by Construction 1 are *universally good* codes.

Fig. 5 shows a 3-regular graph $R_{6,3}$ and the corresponding universally good FR code obtained by Construction 1 for the DSS with $n = 6$ and $d = 3$.

V. Fractional Repetition Codes with $\rho > 2$

Practical systems require the repetition degree to be at least 3 [18], and the previous construction based on regular graphs cannot be generalized to this case. We present here two new constructions of FR codes with $\rho > 2$ based on a combinatorial structure known as *Steiner system* that can be thought of as a generalization of the projective plane of Example 2.

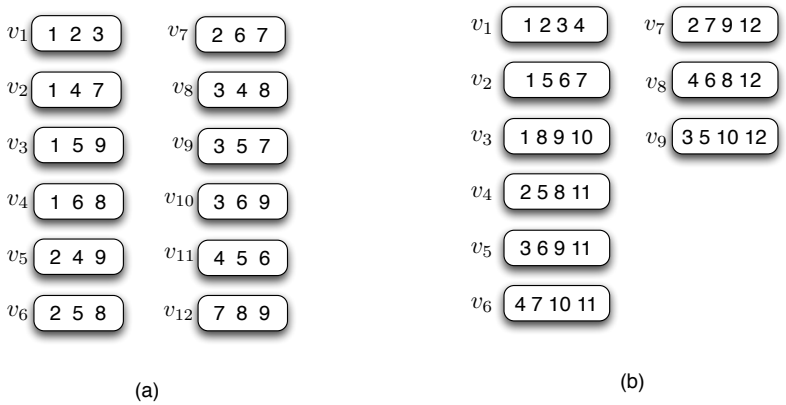

Fig. 6. (a) FR code with $\rho = 4$ for a DSS with $n = 12$ and $d = 3$ derived from the Steiner system S(2,3,9) using Construction 2. (b) FR code with $\rho = 3$ for a DSS with $n = 9$ and $d = 4$ derived from the same Steiner system using Construction 3.

A. Steiner Systems

We start by giving a definition of a Steiner system.

Definition 7 (Steiner System): A Steiner system $S(t, \alpha, v)$ is a collection of subsets, called blocks, $B_1, \ldots, B_b$, of size α of a set $\mathcal{V}$ containing v elements, called points, with the property that any subset of t points is contained in *exactly* one block.

It can be shown that in a Steiner system every point belongs to exactly the same number of blocks denoted r [15, p. 60]. One way to guarantee the achievability of the capacity in (1) is to require that nodes do not share more than one packet. For this reason, we will be mostly interested in Steiner systems with $t = 2$. Simple counting arguments give the following two well-known properties of a Steiner system $S(2, \alpha, v)$ [15, Chap. 2].

Proposition 8: The parameters b and r of a Steiner system $S(2, \alpha, v)$ are given by:

$$b\alpha = vr, \tag{4}$$

$$v - 1 = r(\alpha - 1). \tag{5}$$

Equation (4) is similar to (3) for FR codes. The Fano plane of Fig. 3(a) is an example of a Steiner system where $\mathcal{V}$ is the set of 7 points and the blocks are the lines (including the circle). The Fano plane is indeed $S(2, 3, 7)$ since there is a single line that goes through any two points. Prop. 8 gives $r = 3$, *i.e.*, each point belongs to exactly 3 lines, and $b = 7$, *i.e.*, the Fano plane contains 7 lines, which can be easily checked on the figure.

For a Steiner system $S(t, \alpha, v)$ to exist, it is necessary that the parameters b and r given in Prop. 8 be integers. Wilson proved in [19] that this condition is also sufficient when v is large enough.

Theorem 9: Given a positive integer α, Steiner systems $S(2, \alpha, v)$ exist for all sufficiently large integers v for which the congruences $vr \equiv 0 \bmod \alpha$ and $v - 1 \equiv 0 \bmod \alpha - 1$, are valid.

B. Code Constructions

We present now two constructions of universally good FR codes derived from Steiner systems. Example 2 suggests the following direct construction.

Construction 2: Given a Steiner system $S(2, \alpha, v)$ with blocks $B_1, \ldots, B_b \subset \mathcal{V} = [v]$, an FR code $\mathcal{C}$ can be obtained by taking $\mathcal{C} = \{B_1, \ldots, B_b\}$. This gives an FR code with $\rho = \frac{v-1}{\alpha-1}$ and $\theta = v$ for a DSS with parameters $n = \frac{v(v-1)}{\alpha(\alpha-1)}$ and $d = \alpha$ as given by Prop. 8.

By definition, any two blocks in $S(t, \alpha, v)$ cannot intersect in more than $t-1$ elements. This implies that in the FR codes obtained by Construction 2, two nodes can have at most one packet in common. Thus, in any collection of k nodes, there are at most $\binom{k}{2}$ repeated packets. Therefore, the obtained FR codes can achieve the capacity C_{MBR} for all $k = 1, \ldots, d$ as stated in the following lemma.

Lemma 10: The FR codes obtained by Construction 2 are universally good.

Construction 2 has the disadvantage that the two important code design parameters, n and ρ do not figure explicitly in the Steiner system parameters. Next, we present a second construction where n and ρ directly determine the Steiner system and where the repair degree d is a fraction of the survivor nodes.

Construction 3: Given a Steiner system $S(2, \alpha, v)$ with blocks $B_1, \ldots, B_b \subset \mathcal{V} = [v]$, an FR code $\mathcal{C} = \{V_1, \ldots, V_n\}$ can be obtained by taking

$$V_i = \{j | i \in B_j\},$$

for $i = 1, \ldots, n$. This gives an FR code with $\rho = \alpha$ and $\theta = \frac{n(n-1)}{\rho(\rho-1)}$ for a DSS with parameters $n = v$ and $d = \frac{n-1}{\rho-1}$ as given by Prop 8.

We refer to the codes obtained by this construction as *Transpose* codes since the role of the blocks and points are reversed. The blocks now correspond to packets and the points to the storage nodes. Therefore, any two nodes have exactly one packet in common. Therefore, we get the following lemma.

Lemma 11: The FR codes obtained by Construction 3 are universally good.

To highlight the difference between these two constructions, we give an example in Figure 6 when they are both applied to the unique Steiner system S(2,3,9) [20, p. 27]. Construction 2 gives an FR code with $\rho = 4$ for a DSS with $n = 12$ and $d = 3$, whereas Construction 3 gives an FR code with $\rho = 3$ for a DSS with $n = 9$ and $d = 4$. Note that these two constructions will give the same FR code (up to relabeling) when applied to projective planes such as the Fano plane of Fig. 3(b).

The previous two constructions assume the existence of the Steiner system with the desired parameters, which is not always true. However, Steiner systems $S(2, \alpha, v)$ are known to exist for small values of α, namely $\alpha = 2, \ldots, 5$, and for any v whenever the integrality conditions given by Th. 9 are satisfied. This result in conjunction with Construction 3 gives the necessary and sufficient conditions for the existence of Transpose codes with low repetition degree.

Corollary 12: Transpose codes with repetition degree $\rho = 2, \ldots, 5$ exist if and only if $n - 1 \equiv 0 \bmod \rho - 1$ and $n(n-1) \equiv 0 \bmod \rho(\rho - 1)$.

The previous corollary implies that for the important practical case of systems with repetition degree $\rho = 3$, universally good FR codes with repair degree $d = \frac{n-1}{2}$ can be obtained by Construction 3 using Steiner systems $S(2, 3, n)$ which exist for all $n \equiv 1, 3 \mod 6$. Steiner systems with $\alpha = 3$, known in the literature as Steiner triple systems, are historically the most investigated systems and explicit constructions, such as Bose and Skolem constructions, exist for all feasible values of n [21].

VI. Capacity under Exact Uncoded Repair

The universally good FR codes constructed in the previous sections are guaranteed to have a rate greater or equal to the capacity C_{MBR} of the system under random access and functional repair. However, there exist cases where FR codes can achieve a storage capacity that exceeds C_{MBR}. For instance, consider the FR code $\mathcal{C} = \{\{1,2,3\},\{4,5,6\},\{7,8,9\},\{1,4,7\},\{2,5,8\},\{3,6,9\}\}$ of repetition degree 2 for the $(6,3,3)$ DSS is depicted in Fig. 7. It can be checked any user contacting 3 nodes observes at least 7 distinct packets. Therefore, this code has a rate $R_{\mathcal{C}}(3) = 7 > C_{MBR} = 6$.

We refer to the maximum file size that a DSS with parameters (n, k, d) can store under exact and uncoded repair as its *Fractional Repetition (FR) capacity* C_{FR} defined as follows:

Definition 13 (Fractional Repetition Capacity): The Fractional Repetition (FR) capacity, denoted by $C_{FR}(k, \rho)$ of a distributed storage system with parameters (n, k, d) is defined, for all ρ satisfying $nd \equiv 0 \bmod \rho$, as

$$C_{FR}(k, \rho) := \max_{\mathcal{C}} R_{\mathcal{C}}(k),$$

where $\mathcal{C}$ is any FR code with repetition degree ρ for an (n, k, d) DSS.

The condition on ρ in the definition above is needed by Prop. 8 to guarantee the existence of an FR code $\mathcal{C}$. Note that this notion of capacity assumes that a packet is an atomic unit of information that cannot be divided, which is usually true in real applications such the one in [18] where packets are of size 64 MB.

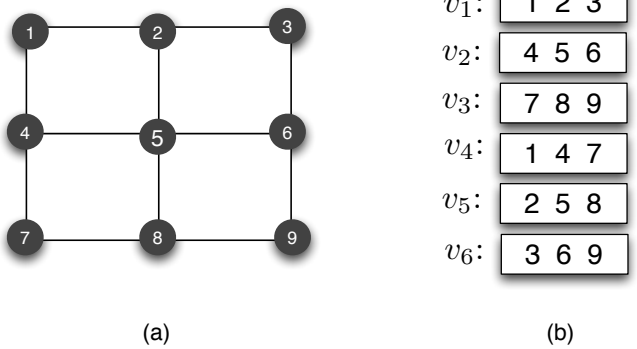

Fig. 7. (a) A 3×3 grid of 9 points and 6 lines. (b) The corresponding FR code achieving a storage capacity exceeding C_{MBR}.

The code constructions of the previous sections imply lower bounds on the FR capacity. Next, we present two upper bounds on C_{FR}. The first is based on an averaging argument and is presented in Lemma 14.

Lemma 14: For a DSS with parameters (n, k, d),

$$C_{FR}(k, \rho) \leq \left\lfloor \frac{nd}{\rho} \left(1 - \frac{\binom{n-\rho}{k}}{\binom{n}{k}} \right) \right\rfloor.$$

Proof: Let $\mathcal{C} = \{V_1, \ldots, V_n\}$ be an FR code with repetition degree ρ, where $V_i \subset [\theta], |V_i| = d$ and $\theta = \frac{nd}{\rho}$ as given by Prop. 5.

Define the set $\mathcal{U}$ as

$$\mathcal{U} := \{U_I = \cup_{i \in I} V_i : I \subset [n], |I| = k\}.$$

The set U_I represents the set of packets observed by a user contacting the nodes in the DSS indexed by the elements in I. We want to show that the term on the right in the inequality is the average cardinality of the sets in $\mathcal{U}$ under uniform distribution. We denote this average by $\overline{U}$. To find $\overline{U}$, we count the following quantity $\sum_{U_I \in \mathcal{U}} |U_I|$ in two ways.

First, we have by definition

$$\sum_{U_I \in \mathcal{U}} |U_I| = \binom{n}{k} \overline{U}.$$

But, each element in $[\theta]$ belongs to exactly $\binom{n}{k} - \binom{n-\rho}{k}$ sets in $\mathcal{U}$. Therefore,

$$\sum_{U_I \in \mathcal{U}} |U_I| = \theta \left(\binom{n}{k} - \binom{n-\rho}{k} \right).$$

The upper bounds follows then from the fact that there must be in $\mathcal{U}$ at least one set of cardinality less that the average. ■

For instance, for the DSS $(7, 3, 3)$, Lem. 14 implies that $R(3, 3) \leq \lfloor 6.2 \rfloor = 6$. Therefore, the FR code of Example 2 is optimal and $C_{FR}(3, 3) = 6$. However, the above upper bound

has the disadvantage of becoming loose for large values of n and k since the FR capacity is by definition a worst case measure.

We also give a second bound on the FR capacity of a DSS which is defined using a recursive function and is tighter than the previous one.

Lemma 15: For a DSS (n, k, d), the FR capacity is upper bounded by the function $g(k)$, *i.e.*, $C_{FR}(k, \rho) \leq g(k)$, where $g(k)$ is defined recursively as

$$g(1) = d, \tag{6}$$

$$g(k+1) = g(k) + d - \left\lceil \frac{\rho g(k) - kd}{n-k} \right\rceil. \tag{7}$$

The proof for this lemma is omitted due to space restrictions and can be found in [22].

VII. Conclusion and Open Problems

We proposed a new class of Exact Minimum-Bandwidth Regenerating (MBR) codes for distributed storage systems characterized by a low complexity *uncoded* repair process. The main component of our construction is a new code that we call Fractional Repetition (FR) code. An FR code with repetition degree ρ guarantees uncoded repair for up to $\rho - 1$ failures. It consists of splitting the data on each node into multiple packets and storing ρ replicas of each on distinct nodes in the system. An additional outer MDS code guarantees that a user contacting a sufficient number of storage nodes will be able to retrieve the stored file.

For single node failures, *i.e.*, $\rho = 2$, we presented a construction of FR codes based on *regular graphs* for all feasible system parameters. For the multiple failures case, *i.e.*, $\rho > 2$, we presented two code constructions based on *Steiner systems*. Of particular importance are the constructed *Transpose* codes where the nodes contacted for repair are just a fraction of the surviving ones. All the obtained codes are guaranteed to achieve the storage capacity under random-access repair. The adopted table-based repair model motivates a new concept of capacity for distributed storage systems, referred to as of Fractional Repetition (FR) capacity, which we studied and derived corresponding bounds.

This work constitutes the first step in the study of Fractional Repetition codes and many important questions remain open. For instance, it is not known whether FR codes with $\rho > 2$ exist for system parameters not covered by our constructions. Moreover, a general expression of the FR capacity is still an open problem, as well as codes that can achieve it.

References

[1] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz, "Maintenance-free global data storage," *IEEE Internet Computing*, pp. 40–49, 2001.

[2] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," in *INFOCOM'07*, 2007.

[3] A. Dimakis, P. Godfrey, Y. Wu, M. Wainright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inform. Theory*, vol. 56, pp. 4539–4551, Sep. 2010.

[4] Y. Wu, A. G. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," in *Proc. Allerton Conference on Control, Computing and Communication*, 2007.

[5] K. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *Allerton Conference on Control, Computing, and Communication*, 2009.

[6] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit and optimal exact-regenerating codes for the minimum-bandwidth point in distributed storage," in *Int. Sym. on Inf. Th. (ISIT'10)*, 2010.

[7] C. Suh and K. Ramchandran, "On the existence of optimal exact-repair MDS codes for distributed storage," tech. rep., 2010.

[8] V. R. Cadambe, S. A. Jafar, and H. Maleki, "Distributed data storage with minimum storage regenerating codes - exact and functional repair are asymptotically equally efficient," in *arXiv:1004.4299v1 [cs.IT]*, 2010.

[9] Y. Wu and A. G. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," in *IEEE Internat. Symp. Inform. Th.*, 2009.

[10] C. Suh and K. Ramchandran, "Exact regeneration codes for distributed storage repair using interference alignment," in *Proc. IEEE Intl Symp. on Information Theory (ISIT)*, 2010.

[11] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Explicit codes minimizing repair bandwidth for distributed storage," in *ITW*, 2010.

[12] V. Venkatesan, "Fast rebuilds in distributed storage systems using network coding," tech. rep., IBM Research GmbH, Zurich Research Laboratory, 2009.

[13] S. Pawar, S. El Rouayheb, and K. Ramchandran, "On secure distributed data storage under repair dynamics," in *Int. Sym. on Inf. Th. (ISIT'10)*, 2010.

[14] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *submitted to Special Issue of the IEEE Trans. on Inf. Th. (Facets of Coding Theory)*, 2010.

[15] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North Holland, June 1988.

[16] N. C. Wormald, "Models of random regular graphs," *London Mathematical Society Lecture Note Series*, vol. 267, pp. 239–298, 1999.

[17] J. Kim and V. H. Vu, "Generating random regular graphs," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, (San Diego, CA, USA), pp. 213–222, 2003.

[18] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in *19th ACM Symposium on Operating Systems Principles*, 2003.

[19] R. M. Wilson, "An existence theory for pairwise balanced designs: III-proof of the existence conjectures," *J. Comb. Theory*, vol. 18A, pp. 71–79, 1975.

[20] C. J. Colbourn and J. H. Denitz, *Handbook of Combinatorial Designs, Second Edition*. Chapman and Hall/CRC, 2006.

[21] C. C. Lindner and C. A. Rodger, *Design Theory, Second Edition*. Chapman and Hall/CRC, 2008.

[22] S. El Rouayheb and K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems (extended version)." under preparation.